

Java mapping for pragmatic programmers



October 8, 2013

Java mapping for pragmatic programmers

Scala Edition

October 8, 2013



Disclaimer

Disclaimer

- I'm not a Scala expert, actually, I do not know Scala at all.

Disclaimer

- I'm not a Scala expert, actually, I do not know Scala at all.
- Lets discuss today not implementations, but ideas staying behind them.

Agenda

1. Parser combinators
2. JAXB plugin
3. Scala Macros
4. Q&A, Discussion

XRools

- Written by Scala experts from Kiev
- XPath like DSL
- Extendable by Scala code
- XML oriented

XRools

Talk is cheap. Show me the code.

XRools Pros

- Scala

XRoos Pros

- Scala
- Non-developers write rules

XRoos Pros

- Scala
- Non-developers write rules
- Dashboard

- 
- ENGINEERING BUSINESS PERFORMANCE

XRoos Cons

- Scala
- Non-developers write rules

XRoops Cons

- Scala
- Non-developers write rules
- No compile-time checks

XRoles Cons

- Scala
- Non-developers write rules
- No compile-time checks
- Uses reflection

XRoles Cons

- Scala
- Non-developers write rules
- No compile-time checks
- Uses reflection
- No real IDE support

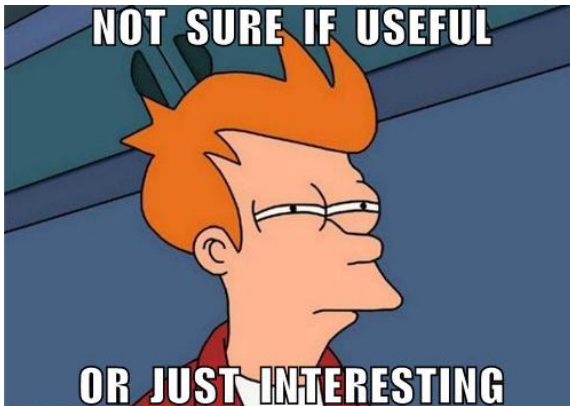
XRoles Cons

- Scala
- Non-developers write rules
- No compile-time checks
- Uses reflection
- No real IDE support
- Functions defined away from its only usage

XRoles Cons

- Scala
- Non-developers write rules
- No compile-time checks
- Uses reflection
- No real IDE support
- Functions defined away from its only usage
- Debugging?

XRoles



JAXB plugin

- Written by Sergey Armensky (idea of Andrey Vytnov?)
- Plain Java (new method added for each property)
- Will be soon in production
- Have nothing to do with Scala

JAXB plugin

Talk is cheap. Show me the code.

JAXB plugin Pros

- Pure Java

JAXB plugin Pros

- Pure Java
- Simple and clean

JAXB plugin Pros

- Pure Java
- Simple and clean
- Simplify some code

JAXB plugin Cons

- Only for JAXB generated classes

JAXB plugin Cons

- Only for JAXB generated classes
- Simplify only some code

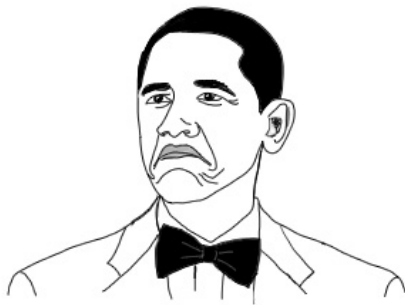
JAXB plugin Cons

- Only for JAXB generated classes
- Simplify only some code
- Not declarative enough

JAXB plugin Cons

- Only for JAXB generated classes
- Simplify only some code
- Not declarative enough
- Not mapper at all

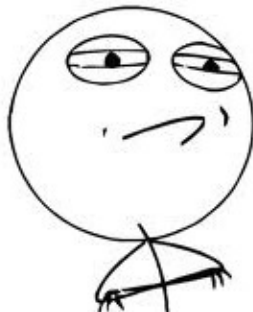
JAXB Plugin



NOT BAD

Can we do better?

Can we do better?



CHALLENGE ACCEPTED

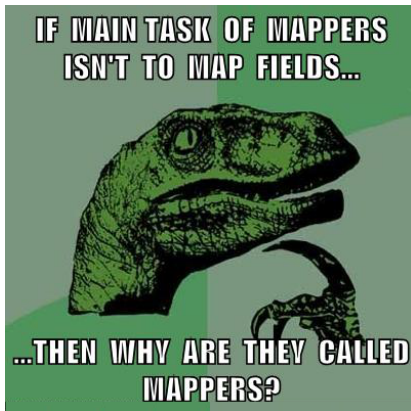
Scala related features

- Implicits of all kinds
- Point-free style
- Scala virtualized
- Macros

Yajom

- Yet Another Java Object Mapper
- Based on implicits and Scala macros
- All reflection is done in compile-time
- Easy to configure, extend or derive
(having no clue about Scala Macros at all)

Philosoraptor



Java with JAXB plugin

```
100 A a = from.getA();
101 if (verySpecial(from)) {
102     B b = defaultB;
103     if (a != null) {
104         B realB = a.getB();
105         if (realB != null)
106             b = realB;
107     }
108     to.c().d().e().setB(b);
109 }
```

Option-based yajom

```
100 yajomOption(to.getC.getD.getE.setB) {
101     val a = from.getA
102     if (verySpecial(from)) {
103         var b = defaultB
104         if (a != null) {
105             val realB = a.getB
106             if (realB != null)
107                 b = realB
108         }
109         Some(b)
110     } else
111         None
112 }
```

Option-based yajom with maybe

```
100 yajomOption(to.getC.getD.getE.setB) {  
101     if (verySpecial(from)) {  
102         val b = maybe(from.getA.getB)  
103         Some(b.getOrElse(defaultB))  
104     } else  
105         None  
106 }
```

Yajom

Talk is cheap. Show me the code.

Yajom Pros

- Scala

- Scala
- Compile-time bulletproof

Yajom Pros

- Scala
- Compile-time bulletproof
- IDE support

Yajom Pros

- Scala
- Compile-time bulletproof
- IDE support
- Works with any classes without any changes

Yajom Pros

- Scala
- Compile-time bulletproof
- IDE support
- Works with any classes without any changes
- No problem with debugging

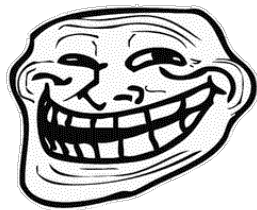
- Scala

Yajom Cons

- Scala
- Maybe not mature enough

Yajom Cons

- Scala
- Maybe not mature enough



Yajom



<https://github.com/gark87/yajom>

Thank You!